Eclipse Plug-in Development Research Project

Report 3

UNIVERSITY AT BUFFALO

July 2, 2015

Authored by: Chern Yee Chua

Eclipse Plug-in Development Research Project

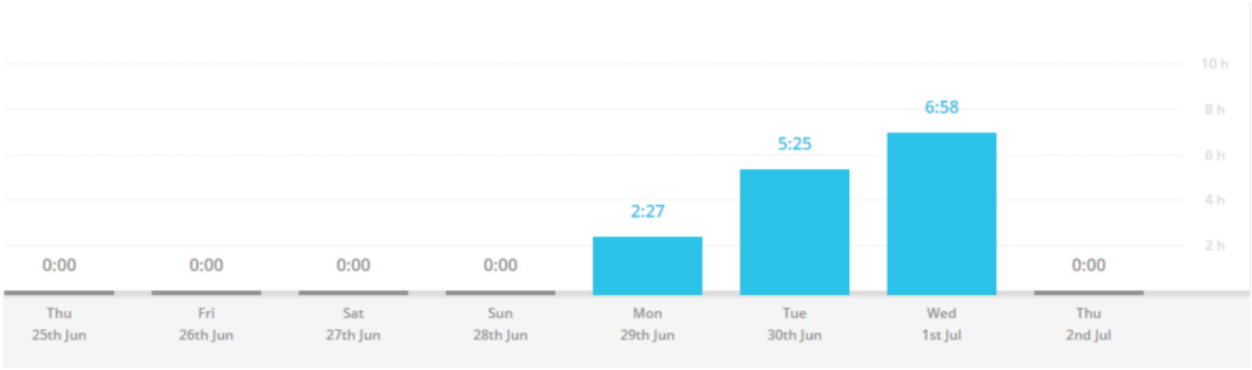
Report 3

Introduction

In this week, I manage to accomplish few things: (1) get more on element changes, (2) write the changes to the file, (3) implement time stamp to keep track of changes

Synopsis

I have been spending more than 10 hours (including report writing) on this project. Below is the summary graph:



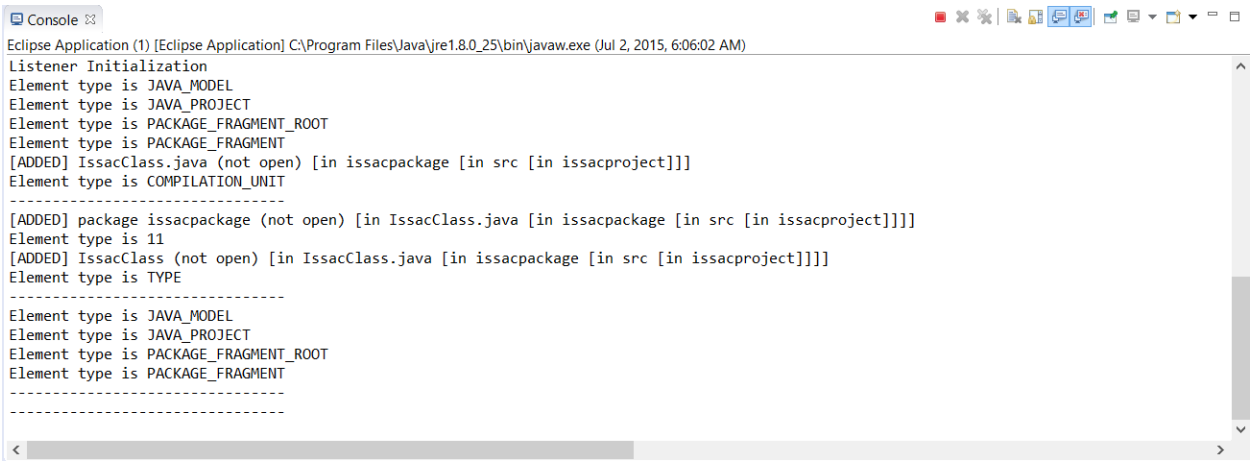
Eclipse Plugin	**14:51**:14
detect more java element	**7:52**:39
reinvent the wheel	**6:58**:35

Content and Walkthrough

This project is still in development stage and there will be a lot more details and functionality added in as time goes by. Below is a simple guide on using this plugin (not fully implemented and tested but just to show how this plugin works and its expectation in the next development):

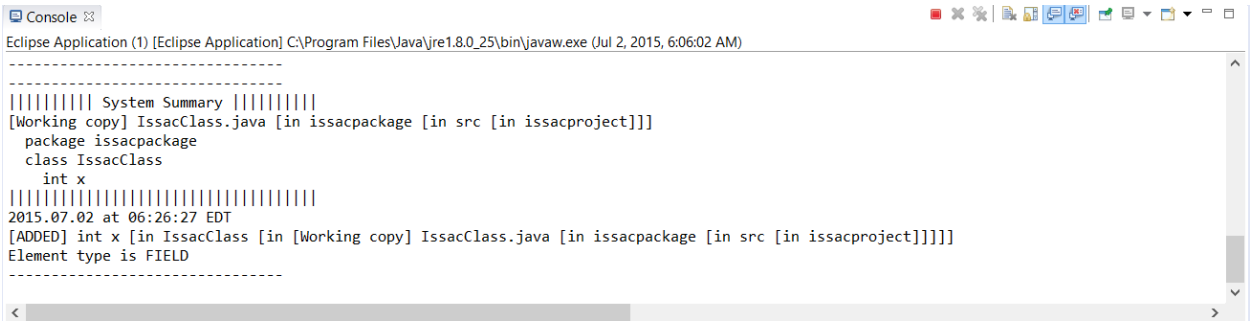
1. Run the plugin project by clicking 
   (In the future, this plugin should be installed on eclipse and run automatically)

2. Create a new Java Project / package / class and etc. will trigger the elementChanged event. Snapshots below show what happen when I create a new class named IssacClass.



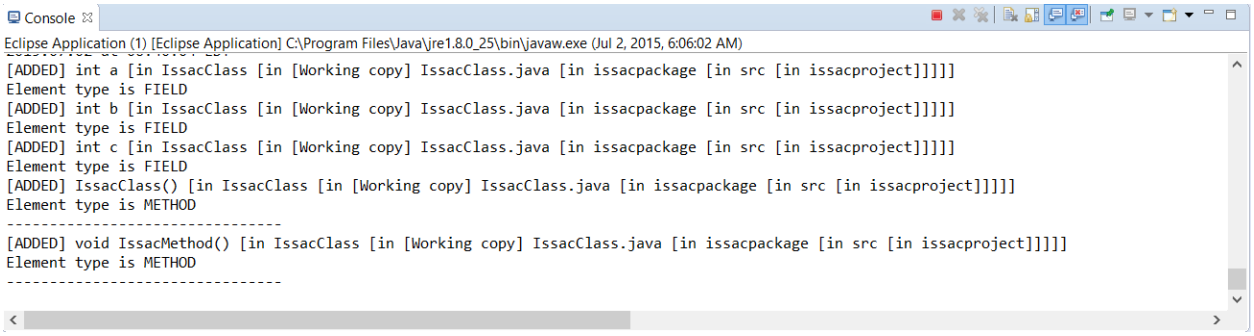
As you can see, those lines with "Element type is" is actually a result of element changes. I use the delta.getElement().getElementType() to detect what type of changes in the event. Apparently, adding a new class will trigger the following items: Java Model, Java Project, Package Fragment Root, and Package Fragment. Those are the delta affected children. In this level, I only detect certain element types for testing purposes.
Next, those "[ADDED]" lines are the elements that are added as part of the element changes. (Fun fact: you can actually use System.out.println on the elements that are not String type. For the example above, returning a delta element gives the important message of the context)

3. Now, let's try adding some variables in the Java class file. I add an instance variable inside IssacClass and this is the output:

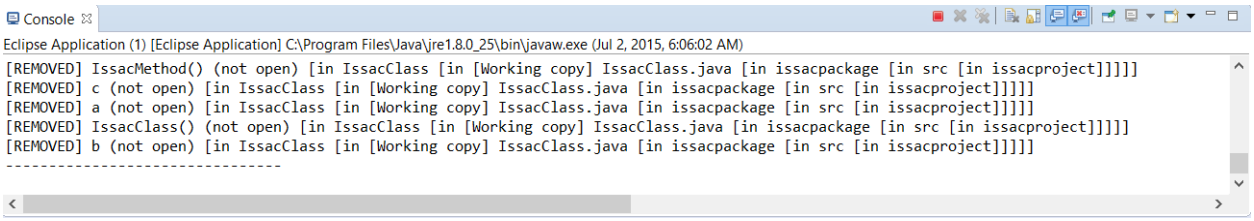


It is marked as "[ADDED]" and its element type is FIELD as you can see from the console. For every minute, the plugin listener will provide a summary on all of the element changes with the little time stamp attached at the end of the summary. The snapshot of the summary will not occur in next minute until the next elementChanged event takes place.

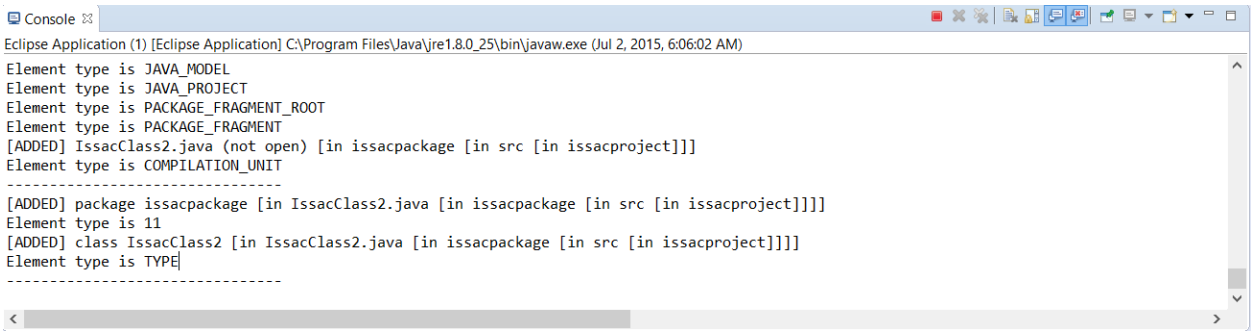
4. Let's play around a little more.



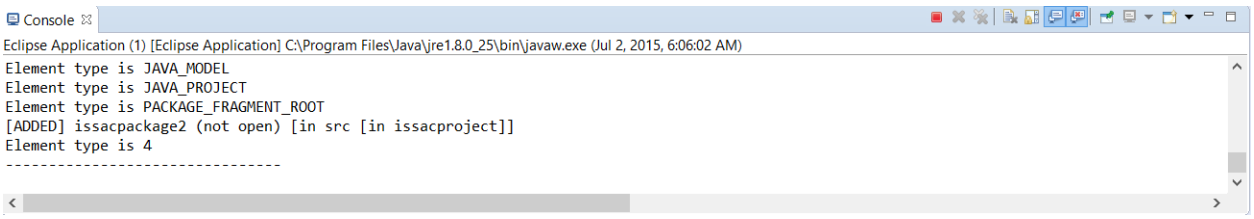
From the snapshot, it is very clear that variables a, b, and c as well as a constructor and a method are added into IssacClass.java in issacpackage, which is inside the source folder of issacproject. Now, let's remove all the elements we added in:



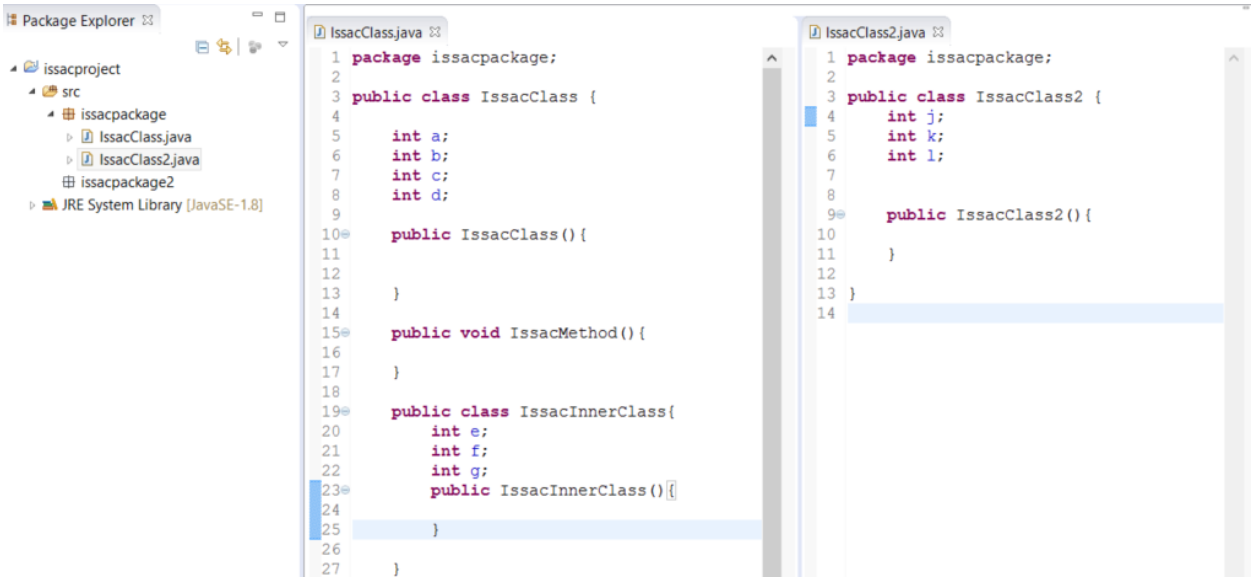
Let's add another java class called IssacClass2:



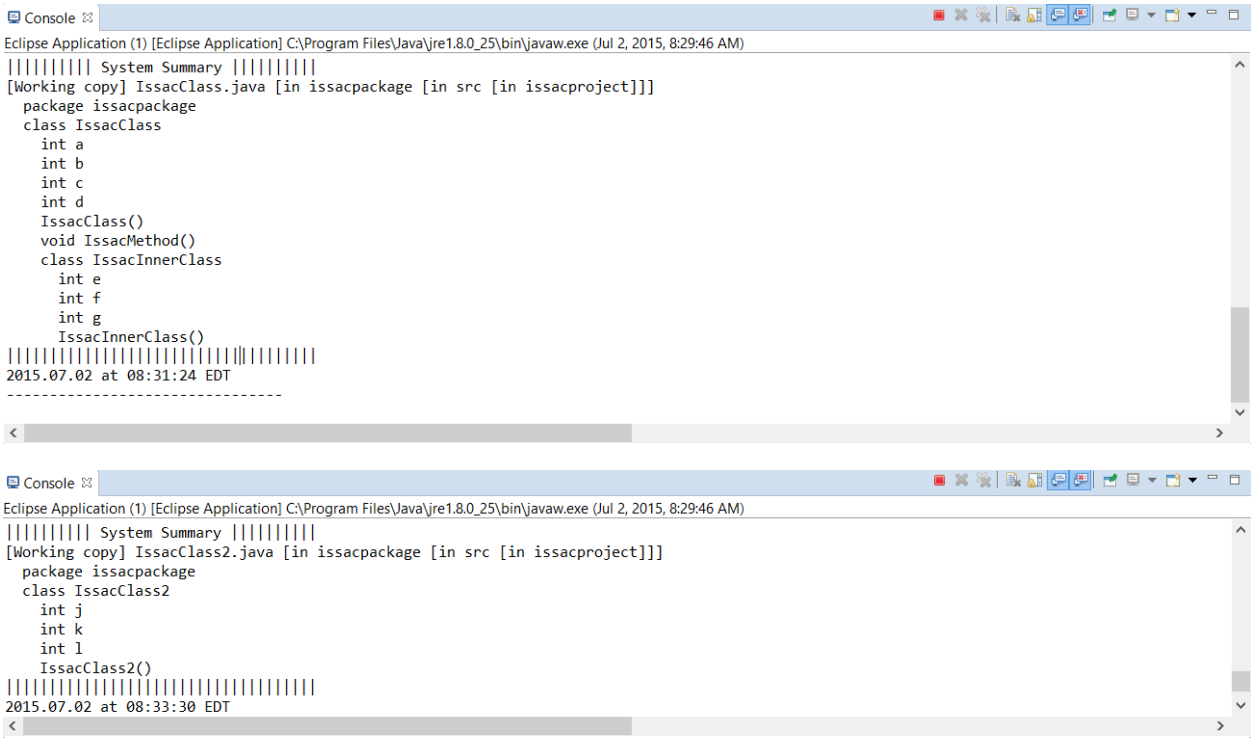
What about a new package called issacpackage2:



By the way, the element type 11 is PACKAGE_DECLARATION and element type 4 is PACKAGE_FRAGMENT. (I forgot to add it in)



Snapshot above shows the classes inside the project and snapshot below shows the summary of the result.



5. When you close the working space, an output file that contained all the history of changes will be created inside eclipse folder.

Problem statements

Unfortunately, I still can't capture the element changes inside the methods. I have tried all the possible ways but still no luck with it. This has been bugging me for some time.

Secondly, I need to work on various specific element changes instead of relying on solely the IJavaElementDelta's print line from console.

To-do lists:

- Categorize the output in a hierarchical view
- Capture changes inside method
- Instead of using console println, output shouldn't be displayed but store without any notification

Conclusion

Throughout the third week of development, I feel like I know more about the Java Model. More specifically, I know how the java elements behave and how to handle them properly. There are a lot of stuff for me to explore and I will be digging more into this. Thank you for your time.